

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

Docket No.: 100203448-2
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Bradley G. Culter

Application No.: 10/685,287

Confirmation No.: 5828

Filed: October 14, 2003

Art Unit: 2192

For: ALLOWING FIRMWARE TO BORROW A
PROCESSOR

Examiner: M. J. Yigdall

APPEAL BRIEF

MS Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

As required under 37 C.F.R. § 41.37(a), this brief is filed within two months of the Notice of Appeal filed in this case on January 17, 2008, and is in furtherance of said Notice of Appeal.

The fees required under 37 C.F.R. § 41.20(b)(2) are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief contains items under the following headings as required by 37 C.F.R. § 41.37 and M.P.E.P. § 1206:

- | | |
|------|-----------------------------------|
| I. | Real Party In Interest |
| II | Related Appeals and Interferences |
| III. | Status of Claims |
| IV. | Status of Amendments |

V.	Summary of Claimed Subject Matter
VI.	Grounds of Rejection to be Reviewed on Appeal
VII.	Argument
VIII.	Claims Appendix
IX.	Evidence Appendix
X.	Related Proceedings Appendix

I. REAL PARTY IN INTEREST

The real party in interest for this appeal is:

Hewlett-Packard Development Company, L.P., a Limited Partnership established under the laws of the State of Texas and having a principal place of business at 20555 S.H. 249, Houston, TX 77070, U.S.A. (hereinafter "HPDC"). HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, CA. The general or managing partner of HPDC is HPQ Holdings, LLC.

II. RELATED APPEALS, INTERFERENCES, AND JUDICIAL PROCEEDINGS

There are no other appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

A. Total Number of Claims in Application

There are 34 claims pending in application.

B. Current Status of Claims

1. Claims canceled: None
2. Claims withdrawn from consideration but not canceled: None
3. Claims pending: 1-34
4. Claims allowed: None
5. Claims rejected: 1-34

C. Claims On Appeal

The claims on appeal are claims 1-34

IV. STATUS OF AMENDMENTS

Applicant did file an Amendment After Final Rejection which was entered by Examiner (see Advisory Action dated December 28, 2007). A complete listing of the current claims is provided in the claims appendix.

V. SUMMARY OF CLAIMED SUBJECT MATTER

A concise summary of subject matter of the independent claims involved in the appeal, which refers to the specification and to the drawings by reference characters, is provided below. All references to the specification and drawings are made by way of example for the convenience of the Board. No limitation on the following claim language is intended.

An embodiment of the invention, as set forth in claim 1, provides a method for changing control of a processor that is in an active state under the control of an operating system to a borrowed state wherein the processor is under control of firmware (Fig 2, item 200). The method comprises: sending a request for a change in control to the operating system (Fig 2, item 202); deciding, by the operating system, whether to grant the request (Fig 2, item 203); placing the processor in a transitional state that is different from the active state, if the request is granted

(Fig 2 item 204); and sending, by the operating system, an interrupt signal to move the processor from the transitional state into the borrowed state (Fig 2 item 208). *See ¶¶ [0006], [0015-0020]* at pg 2 lines 21-28, and pg 5 line 17- pg 7 line 13.

An embodiment of the invention, as set forth in claim 15, provides a computer system (Fig 4, item 400). The computer system comprises: a processor for executing code; an operating system that has control of the processor during an active state (Fig 4, item 402); and a firmware layer that has control of the processor during a borrowed state (Fig 4, item 411); wherein the operating system decides whether to place the processor in the borrowed state and sends an interrupt signal to move the processor into the borrowed state. *See ¶¶ [0029-0031]* at pg 11 line 5 – pg 12 line 5.

An embodiment of the invention, as set forth in claim 31, provides a computer system (Fig 4, item 400) that has a processor for executing code, an operating system that has control of the processor during an active state (Fig 4, item 402); and a firmware layer that has control of the processor during a borrowed state (Fig 4, item 411), the system comprising: means for forming a request to change control of the processor from the active state to a borrowed state (Fig 4, item 411 412); means, operative by the operating system, for deciding whether to grant the request (Fig 4, item 405); means for moving the processor from the active state to the borrowed state (Fig 4, item 406). *See ¶¶ [0029-0036]* at pg 11 line 5 – pg 14 line 2.

Claim 32 is dependent on claim 31 and further comprises a means for returning the processor to the active state upon completion of a task (Fig 4 406, 407, and 422). *See ¶¶ [0035-0037]* pg 13 line 12 – pg 14 line 9.

An embodiment of the invention, as set forth in claim 33, provides a computer readable medium having computer program logic recorded thereon for changing control of a processor that is in an active state under the control of an operating system to a borrowed state wherein the processor is under control of firmware, comprising: logic for forming a request for a change in control to the operating system (Fig 2, item 202); logic for deciding, by the operating system,

whether to grant the request (Fig 2, item 203); logic for placing the processor in a transitional state that is different from the active state, if the request is granted (Fig 2 item 204); and logic for moving the processor from the transitional state into the borrowed state (Fig 2 item 208). *See* ¶¶ [0006], [0015-0020] at pg 2 lines 21-28, and pg 5 line 17- pg 7 line 13.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. First Ground of Rejection

Claims 1-6, 8, 10-12, 14-20, 23-26, and 28-34 are rejected under 35 U.S.C. § 102(e) as being anticipated by O'Shea (U.S. Application 2004/0128568).

B. Second Ground of Rejection

Claims 7, 9, 21, and 22 are rejected under 35 U.S.C. § 103(a) as being unpatentable over O'Shea in view of the article entitled "Advanced Configuration and Power Interface Specification", revision 2.0b, dated 11 October 2002, (hereinafter the ACPI specification).

C. Third Ground of Rejection

Claims 13 and 27 are rejected under 35 U.S.C. § 103(a) as being unpatentable over O'Shea in view of the article entitled "Itanium Processor Family Abstraction Layer Specification", Intel Document No. 245359-005, July 2001 (hereinafter SAL specification).

VII. ARGUMENT

Appellant respectfully traverses the outstanding rejections of the pending claims, and requests that the Board reverse the outstanding rejections in light of the remarks contained herein. The claims do not stand or fall together. Instead, Appellant presents separate arguments for various independent and dependent claims. Each of these arguments is separately argued below and presented with separate headings and sub-heading as required by 37 C.F.R. § 41.37(c)(1)(vii).

A. First Ground of Rejection

Claims 1-6, 8, 10-12, 14-20, 23-26, and 28-34 are rejected under 35 U.S.C. § 102(e) as being anticipated by O'Shea. "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference," *Verdegaal Bros. v. Union Oil Co. of Cal.*, 814 F.2d 628, 631, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987). Because the O'Shea reference fails to teach each and every claim element in the present application, Appellant respectfully submits that the above rejections are improper.

1. Claims 1-6, 8, 10-12, and 14

Claim 1 recites "deciding, by the operating system, whether to grant the request...and sending, by the operating system, an interrupt signal to move the processor from the transitional state into the borrowed state." Appellant has persistently argued that nothing in O'Shea teaches an operating system making a decision on whether to grant a request for a change in control. Examiner responds by stating that the decision in O'Shea is made based on whether "further execution of the code of the operating system" is taking place. This however, only shows that the decision is made based on whether the operating system is executing code. It does not show that the operating system is making the decision.

In the December 28, 2007 advisory action, Examiner argues that steps 350 and 352 of Fig 3 of O'Shea teach that the operating system is, in part, making the decision. However, the teachings of these steps, instead, further supports Appellant's arguments. Figure 3 of O'Shea shows a flowchart which illustrates the functionality of the usurping code. In step 350, the usurping code checks to see whether operating system code has been executed during the time where the usurping code was making preparations to usurp control of the processor. Clearly, the operating system is not making any decision in this step. It is merely being monitored. In step 352, O'Shea teaches that the usurping code can abort the usurping process in the event that the operating system is executing code. Again, there is no teaching of the operating system making these decisions; rather the decision is being made based on a state of the operating system. It appears that the "usurping code" makes the decision based on its monitoring of the operating

system's activities. And, the usurping code does not appear to be part of the OS, but instead, separate code that attempts to monitor the OS activities and decide whether to usurp control of the processor. Thus, the external usurping code makes the decision regarding whether to usurp control of the processor from the OS, rather than allowing the operating system to itself make this decision.

Further, the very first lines of O'Shea make clear that the power management control functionality of the computer system is used to usurp control. (See O'Shea [0001] lines 1-3). Hence, as shown above, the power management controls in O'Shea monitor the computer system to determine whether an operating system is executing code, and interrupt decisions are made in response to the monitoring (as shown in steps 350/351 of O'Shea). Thus, in O'Shea, the operating system does not make the usurping decision relating to handing over control of the processor.

To further illustrate Applicant's point that O'Shea does not use the operating system in decision making, it is noted that there is a large technical difference between "usurping" control in O'Shea, and "borrowing" control of the processor as discussed in the present invention. One with skill in the art would recognize that operating systems will consider usurped processors to have failed in the event that the OS attempts to use the processor after it has been usurped. Additionally, as a result, if the processor comes back on line after being usurped, an OS will need to have some further protocol to follow in order to re-recognize the presence of the processor. The present invention's use of borrowing overcomes this shortcoming.

Conversely, where an OS makes the decision to allow a processor to be "borrowed", the confusion is eliminated. Thus, the OS remains aware of the processor's existence and expects that the processor control will be returned to the OS. It is clear that the techniques employed in the O'Shea reference usurp control of the computer system when the OS is not executing code (see abstract; [0014]), as opposed to the present invention which allows for borrowing the processor with permission of the OS. In fact, O'Shea explicitly states that control is "stolen away" from the operating system (§ [0026] line 20).

Examiner asserts that when the terms are given their broadest reasonable meaning, there is no distinguishable difference. Appellant disagrees. Even if Appellant were to ignore how one skilled in the art would view these terms, the plain meaning of the term “borrowed” connotes that the party being borrowed from has an understanding that the borrowed item will be returned. Whereas the plain meaning of the term usurped connotes that the item is stolen, and thus, eliminates the possibility of the party being taken from to have any decision in the matter and imparts no expectation of the usurped item being returned. (The American Heritage Dictionary defines “borrowed” as: To obtain or receive (something) on loan with the promise or understanding of returning it or its equivalent; “usurped” is defined as: to seize and hold by force or without legal authority).

Accordingly, the O’Shea reference fails to teach “deciding, by the operating system, whether to grant the request...and sending, by the operating system, an interrupt signal to move the processor from the transitional state into the borrowed state” as required by independent claim 1. Therefore, Appellant submits that the present rejection is improper and should be overturned.

Claims 2-6, 8, 10-12, and 14 depend, either directly or indirectly, from claim 1 and thus, inherits the limitations of claim 1. As a result, claims 2-6, 8, 10-12, and 14 are allowable for at least the reasons set forth above with respect to independent claim 1. Appellant further asserts that there are many limitations contained in the dependent claims that are patentable in their own right.

2. Claims 15-20, 23-26, and 28-30

Claim 15 defines a computer system wherein the operating system decides whether to place the processor in the borrowed state and sends an interrupt signal to move the processor into the borrowed state. O’Shea does not disclose at least these limitations. As discussed in detail with claim 1, O’Shea fails to teach the OS deciding weather to move the processor into a borrowed state. The Office Action relies on paragraph [0043], lines 1-19 of O’Shea as teaching

these limitations. However, this portion of O'Shea does not discuss having the operating system decide whether to place the processor in the borrowed state. This paragraph allows for code to be executed by the operating system before entry into the lower power state if the code exists. Thus, this paragraph does not discuss having the operating system make any decision. Thus, O'Shea does not teach all of the claimed limitations. Therefore, the Appellant respectfully asserts that for the above reasons claim 15 is patentable over the 35 U.S.C. § 102 rejection of record, and thus the rejection should be overturned.

Claims 16-20, 23-26, and 28-30 depend, either directly or indirectly, from claim 15 and thus, inherits the limitations of claim 15. As a result, claims 16-20, 23-26, and 28-30 are allowable for at least the reasons set forth above with respect to independent claim 15. Appellant further asserts that there are many limitations contained in the dependent claims that are patentable in their own right. For example, claim 28 recites "wherein the operating system decides without involvement of the firmware whether to place the processor in the borrowed state." Examiner cites paragraph 0043 of O'Shea as teaching this limitation. However, as asserted above, there is no teaching in this portion of the operating system having any part in the decision making. Instead, it is the usurping code of O'Shea that is making the decision. It is clear that there is no teaching that the operating system is making the entire decision without involvement of the firmware.

3. Claims 31-32

Claim 31 defines a computer system that has a processor for executing code, and includes means, operative by the operating system, for deciding whether to grant the request. The Office Action relies on paragraph [0043], lines 1-19 of O'Shea as teaching these limitations. However, this portion of O'Shea does not discuss having a means that is operable by the operating system for deciding whether to grant the request to change control. This paragraph allows for code to be executed by the operating system before entry into the lower power state if the code exists. Thus, this paragraph does not discuss having the any means operable by the operating system to make any decision. Thus, O'Shea does not teach all of the claimed limitations. Therefore, the

Appellant respectfully asserts that for the above reasons claim 31 is patentable over the 35 U.S.C. § 102 rejection of record, and thus the rejection should be overturned.

Claim 32 depends from claim 31 and thus, inherits the limitations of claim 31. As a result, claim 32 is allowable for at least the reasons set forth above with respect to independent claim 31.

4. Claims 33-34

Claim 33 defines a computer readable medium having computer program logic recorded thereon for changing control of a processor that is in an active state under the control of an operating system to a borrowed state wherein the processor is under control of firmware. The medium includes logic for deciding, by the operating system, whether to grant the request. The Office Action relies on paragraph [0043], lines 1-19 of O'Shea as teaching these limitations. However, this portion of O'Shea does not discuss having logic for deciding, by the operating system, whether to grant the request for a change in control. This paragraph allows for code to be executed by the operating system before entry into the lower power state if the code exists. Thus, this paragraph does not discuss having any logic for deciding by the operating system to make any decision. Thus, O'Shea does not teach all of the claimed limitations. Therefore, the Appellant respectfully asserts that for the above reasons claim 33 is patentable over the 35 U.S.C. § 102 rejection of record, and thus the rejection should be overturned..

Claim 34 depends from claim 33 and thus, inherits the limitations of claim 33. As a result, claim 34 is allowable for at least the reasons set forth above with respect to independent claim 33.

B. Second Ground of Rejection

Claims 7, 9, 21, and 22 are rejected under 35 U.S.C. § 103(a) as being unpatentable over O'Shea in view of the ACPI specification. To establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490

F.2d 981, 180 U.S.P.Q. 580 (C.C.P.A. 1974). Because O'Shea fails to teach multiple claim limitations as asserted by the Examiner, and the ACPI specification does not remedy these deficiencies, Appellant respectfully submits that the above rejections are improper.

1. Claims 7, 9, 21, and 22

Claims 7, 9, 21, and 22 depend, either directly or indirectly, from independent claims 1 and 15 and thus, inherits the limitations of claims 1 and 15 respectively. As shown above, O'Shea does not teach one or more limitations contained in claims 1 and 15. Further, the ACPI specification does not cure these deficiencies. As a result, claims 7, 9, 21, and 22 are allowable for at least the reasons set forth above with respect to independent claim 15.

C. Third Ground of Rejection

Claims 13 and 27 are rejected under 35 U.S.C. § 103(a) as being unpatentable over O'Shea in view of the SAL specification. To establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 U.S.P.Q. 580 (C.C.P.A. 1974). Because O'Shea fails to teach multiple claim limitations as asserted by the Examiner, and the SAL specification does not remedy these deficiencies, Appellant respectfully submits that the above rejections are improper.

1. Claims 13 and 27

Claims 13 and 27 depend directly from independent claims 1 and 15 and thus, inherits the limitations of claims 1 and 15 respectively. As shown above, O'Shea does not teach one or more limitations contained in claims 1 and 15. Further, the SAL specification does not cure these deficiencies. As a result, claims 13 and 27 are allowable for at least the reasons set forth above with respect to independent claim 15.

In view of the above, Appellant requests that the Board overturn the outstanding rejections of claims 1-34. Attached hereto are a Claims Appendix, Evidence Appendix, and Related Proceedings Appendix. As noted in the attached Evidence Appendix, no evidence

Application No.: 10/685,287

Docket No.: 100203448-2

pursuant to §§ 1.130, 1.131, or 1.132 or entered by or relied upon by the Examiner is being submitted. Also, no related appeals are identified in Section II above, and thus as noted by the Related Proceedings Appendix, no decisions in any such related proceedings are provided.

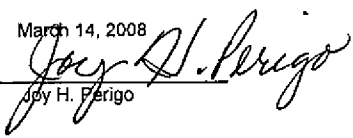
Dated: March 14, 2008

Respectfully submitted,

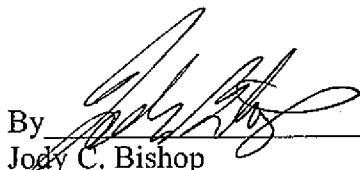
I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being transmitted via the Office electronic filing system in accordance with § 1.6(a)(4).

Dated: March 14, 2008

Signed:


Joy H. Perigo

By


Jody C. Bishop

Registration No.: 44,034

Attorney for Applicant

(214) 855-8007

VIII. CLAIMS APPENDIX

1. A method for changing control of a processor that is in an active state under the control of an operating system to a borrowed state wherein the processor is under control of firmware, comprising:

sending a request for a change in control to the operating system;
deciding, by the operating system, whether to grant the request;
placing the processor in a transitional state that is different from the active state, if the request is granted; and

sending, by the operating system, an interrupt signal to move the processor from the transitional state into the borrowed state.

2. The method of claim 1, further comprising:

maintaining the processor in the active state, if the request is denied.

3. The method of claim 2, further comprising:

re-sending the request, if the request is denied.

4. The method of claim 1, further comprising:

operating the processor in the borrowed state until completion of a task;
placing the processor in another transitional state; and
returning the processor to the active state.

5. The method of claim 4, wherein the task is handling a problem that cannot be handled by the operating system.

6. The method of claim 1, wherein the processor is one processor of a plurality of processors.

7. The method of claim 6, wherein the request specifies the processor.

8. The method of claim 1, wherein a duration of the borrowed state is unbounded.

9. The method of claim 1, wherein the request is an notify command referencing the processor.

10. The method of claim 1, further comprising:
invoking the request by one of the firmware and an application.

11. The method of claim 1, further comprising:
using a general purpose event to cause formation of the request.

12. The method of claim 1, further comprising:
placing the operating system into hibernation while the processor is in the borrowed state.

13. The method of claim 1, wherein the processor is an Itanium® Processor Family chip, and the firmware is system abstraction layer firmware.

14. The method of claim 1, wherein the deciding comprises:
deciding, by the operating system without involvement of the firmware, whether to grant the request.

15. A computer system comprising:
a processor for executing code;
an operating system that has control of the processor during an active state; and
a firmware layer that has control of the processor during a borrowed state;
wherein the operating system decides whether to place the processor in the borrowed state and sends an interrupt signal to move the processor into the borrowed state.

16. The system of claim 15, wherein if the operating system decides not to place the processor in the borrowed state, the operating system maintains control of the processor.

17. The system of claim 15, wherein the firmware maintains control of the processor in the borrowed state until completion of a task, and then returns control of the processor to the operating system.

18. The system of claim 17, wherein the task involves a problem that cannot be handled by the operating system.

19. The system of claim 15, wherein the processor is one processor of a plurality of processors.

20. The system of claim 15, further comprising:
a requesting entity invokes a request that the processor be placed in the borrowed state.

21. The system of claim 20, wherein the processor is one processor of a plurality of processors and the request specifies the processor.

22. The system of claim 20, wherein the request is an notify command referencing the processor.

23. The system of claim 20, wherein the requesting entity is one of:
the firmware, an application, and platform hardware.

24. The system of claim 20, wherein a general purpose event is used to cause formation of the request.

25. The system of claim 15, wherein a duration of the borrowed state is unbounded.

26. The system of claim 15, wherein the operating system is placed into hibernation while the processor is in the borrowed state.

27. The system of claim 15, wherein the processor is an Itanium® Processor Family chip, and the firmware is system abstraction layer firmware.

28. The system of claim 15, wherein the operating system decides without involvement of the firmware whether to place the processor in the borrowed state.

29. The system of claim 15, wherein the operating system acts as a proxy to deliver a message to the firmware layer.

30. The system of claim 15, wherein a modification of the firmware does not require a modification to the operating system.

31. A computer system that has a processor for executing code,
an operating system that has control of the processor during an active state; and
a firmware layer that has control of the processor during a borrowed state, the system comprising:

means for forming a request to change control of the processor from the active state to a borrowed state;

means, operative by the operating system, for deciding whether to grant the request;

means for moving the processor from the active state to the borrowed state.

32. The computer system of claim 31, further comprising:

means for returning the processor to the active state upon completion of a task.

33. A computer readable medium having computer program logic recorded thereon for changing control of a processor that is in an active state under the control of an operating system to a borrowed state wherein the processor is under control of firmware, comprising:

logic for forming a request for a change in control to the operating system;

logic for deciding, by the operating system, whether to grant the request;

logic for placing the processor in a transitional state that is different from the active state, if the request is granted; and

logic for moving the processor from the transitional state into the borrowed state.

34. The computer system of claim 33, further comprising:

logic for returning the processor to the active state upon completion of a task.

IX. EVIDENCE APPENDIX

No evidence pursuant to §§ 1.130, 1.131, or 1.132 or entered by or relied upon by the examiner is being submitted.

X. RELATED PROCEEDINGS APPENDIX

No related proceedings are referenced in II. above, hence copies of decisions in related proceedings are not provided.